

On the Security of the Cha-Ko-Lee-Han-Cheon Braid Group Public-key Cryptosystem

Milton M. Chowdhury

1. Abstract

We show that a number of cryptographic protocols using non-commutative semigroups including the Cha-Ko-Lee-Han-Cheon braid group cryptosystem have security based on the MSCSP. We give two algorithms to solve the DP using the MSCSP.

2. Introduction

At the CRYPTO 2000 conference the seminal KLCHKP (Ko-Lee-Cheon-Han-Kang-Park) braid group public-key cryptosystem was published see [2]. An updated version of the KLCHKP cryptosystem which is the CKLHC (Cha-Ko-Lee-Han-Cheon) braid group cryptosystem was introduced at ASIACRYPT 2001 conference [10] the claim of the authors was the updated cryptosystem is based on the DH-DP (Diffie-Hellman Decomposition Problem). We show that the KLCHKP and CKLHC cryptosystems are based on the MSCSP and it has been assumed for several years the security of these cryptosystems are based on the DH-CP and DH-DP respectively, we also show the related cryptosystems may be based on the MSCSP and hence give a new way to break the KLCHKP and CKLHC cryptosystems and the related cryptosystems for some parameters. It has been shown there is a linear algebraic attack on the KLCHKP and CKLHC cryptosystems but our attack is more practical.

3. Hard Problems in Non-Abelian Groups

Definition-The MSCSP (multiple simultaneous conjugacy search problem) [3] is find elements $g \in G$ such that $y_i = gx_i g^{-1}$, given the publicly known information: G is a group, $x_i, y_i \in G$ with $x_i, y_i = ax_i a^{-1}$, $1 \leq i \leq u$, with the secret element $a \in G$.

Definition-The CSP [3] can be defined as the MSCSP with $u = 1$.

Notation-We refer an example of the MSCSP as $((x_1, x_2, \dots, x_u), (y_1, y_2, \dots, y_u))$ with solution g .

The **DP** (Decomposition Problem) [6] is defined as follows.

Public Information: G is a semigroup, A is a subset of G . $x, y \in G$ with $y = axb$.

Secret information: $a, b \in A$.

Objective: find elements $f, g \in A$ such that $fxg = y$.

The definition of the *DP* above generalises the definition of a less general version of the *DP* given in [8], [3] and [7]. The less general version only differs from the above definition of *DP* because G is a group and A is a subgroup. In

our notation in all of this paper we omit the binary operation $*$ when writing products so for example $f * x * g$ is understood to mean fxg . We require that $*$ is efficiently computable.

The **CSP** (Conjugacy Search Problem) [1], [3] is defined as follows.

Public Information: G is a group. $x, y \in G$ with $y = f^{-1}xf$.

Secret Information: $f \in G$.

Objective: find an element $g \in G$ such that $g^{-1}xg = y$.

Notation-We refer an example of the CSP as (x, y) with solution g .

The **DH-DP** (Diffie-Hellman Decomposition Problem) [8], [3] is defined as follows.

Public Information: G is a group. A, B are subgroups of G with $[A, B] = 1$. $x, y_a, y_b \in G$ with $y_a = axb$, $y_b = cxd$.

Secret Information: $a, b \in A$, $c, d \in B$.

Objective: find the element $cy_ad (= ay_bb = acx_b d)$.

The **DH-CP** (Diffie-Hellman Conjugacy Problem) is the specialisation of the DH-DP [8] with $a = b^{-1}$ and $c = d^{-1}$.

We now re-define the DP and DH-DP above as used in our key agreement protocol given in [12]. In the rest of this paper below the DP and DH-DP will mean their re-definitions.

The re-definition of the **DP** is as follows.

Public Information: G is a semigroup. A, B are subsets of G . $x, y \in G$ with $y = axb$.

Secret Information: $a \in A$, $b \in B$.

Objective: find elements $f \in A$, $g \in B$ such that $fxg = y$.

The re-definition of the **DH-DP** is as follows.

Public information: G is a semigroup. A, B, C, D are subsets of G . $x, y_a, y_b \in G$ with $y_a = axb$, $y_b = cxd$.

Secret Information: $a \in A$, $b \in B$, $c \in C$, $d \in D$. $[A, C] = 1$, $[B, D] = 1$

Objective: find the element $cy_ad (= ay_bb = acx_b d)$.

The EDL problem is to decide if the discrete logarithm of two elements in an abelian group are the same [9]. The **EDL** type problem is as follows [9].

Public information: G is a group. $a, b, y_a, y_b \in G$ with $y_a = uav$, $y_b = wbx$.

Secret Information: $u, v, w, x \in G$.

Objective: Decide if $F_a(y_a) \cap F_b(y_b) \neq \emptyset$. Where $F_\beta(\alpha) = \{(a, b) \in B_n \times B_n : \alpha = a\beta b\}$.

We redefine the EDL problem more generally as follows.

Public information: G is a Semigroup. A, B, C, D are subsets of G . $a, b, y_a, y_b \in G$ with $y_a = uav$, $y_b = wbx$.

Secret Information: $u \in A$, $v \in B$, $w \in C$, $x \in D$.

Objective: Decide if $F_a(y_a) \cap F_b(y_b) \neq \emptyset$. Where $F_\beta(\alpha) = \{(a, b) \in B_n \times B_n : \alpha = a\beta b\}$.

4. Key Agreement Protocol Using Non-Commutative Semigroups

In [12] we introduced a key agreement protocol and a variant of it which we briefly describe below.

- Phase 0. Initial setup

i) G is chosen and is publicly known.

A first method to select the parameters is to select publicly known subsets or privately known subsets L_A, L_B, R_A, R_B and Z of G are chosen for which either property a) below is true or property b) below is true. Let $z \in Z$ with z the publicly known element which is the value of x in the definition of the DH-DP used in the example of the DH-DP in our new authentication scheme.

Following [7] let $g \in G$ for G a group, $C_G(g)$ is the centraliser of g , we sketch the modifications to the authentication scheme (and these apply to the key agreement protocol described below) to give two further methods to select the subgroups as follows. Publicly known subsets or privately known L_A, L_B, R_A, R_B and Z of G are chosen for which either property a) below is true or property b) below for the second and third methods below.

The second method to select the subgroups is A chooses $(a_1, a_2) \in G \times G$ and publishes the subgroups as a set of generators of the centralisers $L_B, R_B, L_B \subseteq C_G(a_1), R_B \subseteq C_G(a_2), L_B = \{\alpha_1, \dots, \alpha_k\}$ etc. B chooses $(b_1, b_2) \in L_B \times R_B$, and hence can compute x below etc. Following [7] there is no explicit indication of where to select a_1 and/or a_2 from. Hence before attempting a length based attack in this case the attacker has to compute the centraliser of L_B, R_B .

So a third method to select the subgroups is

A chooses $L_A = G, a_1 \in G$, and publishes $L_B \subseteq C_G(a_1), L_B = \{\alpha_1, \dots, \alpha_k\}$,

B chooses $L_B = G, b_2 \in G$, and publishes $R_A \subseteq C_G(b_2), R_A = \{\beta_1, \dots, \beta_k\}$,

Hence A chooses $(a_1, a_2) \in G \times C_G(b_2)$ and publishes the subgroup(s) as a set of generators of the centralisers B chooses $(b_1, b_2) \in C_G(a_1) \times G$, and hence can compute x etc. Again there is no explicit indication of where to select a_1 and/or b_2 from. Hence before attempting a length based attack in this case the attacker has to compute the centraliser of L_B and/or R_B .

a) If $z \neq e$ we require the following conditions

$$\begin{aligned} [L_A, L_B] &= 1, & [R_A, R_B] &= 1, \\ [L_B, Z] &\neq 1, & [L_A, Z] &\neq 1, \\ [R_B, Z] &\neq 1, & [R_A, Z] &\neq 1, \\ [L_A, R_A] &\neq 1, & [L_B, R_B] &\neq 1. \end{aligned} \tag{2}$$

All the above conditions for $z \neq e$ can arise by generalising from properties of subgroups used in the SDG or CKLHC schemes for example the second and third conditions in (2) arise from the observations that in general $[LB_n, B_n] \neq 1, [LB_n, UB_n] = 1$.

b) If $z = e$ we require the following conditions

$$\begin{aligned} [L_A, L_B] &= 1, & [R_A, R_B] &= 1, \\ [L_A, R_A] &\neq 1, & [L_B, R_B] &\neq 1, \\ [L_B, R_A] &\neq 1, & [L_A, R_B] &\neq 1. \end{aligned} \tag{3}$$

- Choose $z \in B_n$.
 - ii) A(lice) chooses a secret braid $a_1 \in L_A$, $a_2 \in R_A$, her private key; she publishes $K_A = a_1 z a_2$; the pair (w, K_A) is the public key.
 - i) B(ob) chooses a secret braid $b_1 \in L_B$, $b_2 \in R_B$, her private key; she publishes $K_B = b_1 z b_2$; the pair (w, K_B) is the public key.
 - iii) A and B can compute the common shared secret key κ as $\kappa = a_1 K_B a_2$ and $\kappa = b_1 K_A b_2$ respectively.
 - i) Choose a public w

h is a fixed collision-free hash function from braids to sequences of 0's and 1's or, possibly, to braids, for which this choice for h . Again the above protocol is considered with the commutativity conditions 2 or 3. Note the braids K_A and K_B are rewritten for example a normal form to make the protocol secure. Full detail are given in [12]. It was shown in [12] that the above key agreement protocol is a generalisation of the key protocols given in [10],[2],[6],[7].

5. The Diffie-Hellman Decomposition Problem is Equivalent to the Multiple Simultaneous Conjugacy Search Problem

In this section we will show that the DH-DP is equivalent to the MSCSP in our key agreement protocol in section 2.1 hence showing the key exchange protocols given in [10],[2],[11] are based on the MSCSP and the key exchange protocols in [6],[7] may be based on the MSCSP. In the braid group there are various algorithms MSCSP can be solved with non-negligible probability such as length based algorithms or the algorithm using ultra summit sets given in [4] so for braid group implementations the algorithms we show are based on the MSCSP should not be used. Our result also applies to the variant key exchange protocol and variant authentication scheme given in [12].

We now introduce the concept of a CE (conjugacy extractor) function which we build our attack upon.

Notation-We define a CE (conjugacy extractor) function to be a function that on input of information from an user and information transmitted in a cryptographic protocol gives as its output a conjugacy equation, by conjugacy equation we mean an instantiation of the CSP. We denote i CE functions as CE_i , or CE if there is just one function involved.

Theorem 1

Solving the DH-DP is equivalent solving the MSCSP assuming y_a and/or y_b are invertible elements in the DH-DP.

Proof

The proof is for when considering commutativity conditions 2 or 3 in the generalised protocol above, when condition 3 are used then $x = e$. Firstly define the CE for a protocol based on the DH-DP as follows

$$\begin{aligned} CE_1(R_I, y_a) &= y_a R_I y_a^{-1} = axb R_I b^{-1} x^{-1} a^{-1} = ax R_I x^{-1} a^{-1}, R_I \in D \\ CE_2(S_I, y_b) &= y_b S_I y_b^{-1} = cxd S_I d^{-1} x^{-1} c^{-1} = cx S_I x^{-1} c^{-1}, S_I \in B \\ CE_3(R'_I, y_a) &= y_a^{-1} R'_I y_a = b^{-1} x^{-1} a^{-1} R'_I axb = b^{-1} x^{-1} R'_I xb, R'_I \in C \\ CE_4(S'_I, y_b) &= y_b^{-1} S'_I y_b = d^{-1} x^{-1} c^{-1} S'_I cxd = d^{-1} x^{-1} S'_I xd, S'_I \in A \end{aligned}$$

R_I is chosen from the subset that commutes with the subset that the secret a in y_a is chosen from. S_I is chosen from the subset that commutes with the subset that the secret b in y_b is chosen from etc. Since all the parameters are known to compute the CE_1, \dots, CE_4 are easily computable. Note it is sufficient for one CE to exist to prove the theorem but we may want to compute more than one CE because their difficulty may vary, for example one of the CE of the protocols [7] can be used in a known length attack. Obviously R_I (in general) does not commute with x (similarly R_I in general does not commute with a when conditions 3 are used) as this would mean an attacker could easily recover the common secret key. So for $1 \leq I \leq u$ this shows that the key agreement protocol in [2] is based on the MSCSP for the secret in either ax or the MSCSP in the secret cx . So a or c can be found by right multiplying by x^{-1} which is publicly known. Hence the protocols in [10], [2], [6], [7], [11] are based on an example of the MSCSP as

$$\begin{aligned} &((R_1, R_2, \dots, R_u), (CE_1(R_1, y_a), CE_1(R_2, y_a), \dots, CE_1(R_u, y_a))) \text{ with solution } ax. \\ &((S_1, S_2, \dots, S_u), (CE_2(S_1, y_b), CE_2(S_2, y_b), \dots, CE_2(S_u, y_b))) \text{ with solution } cx. \\ &((R'_1, R'_2, \dots, R'_u), (CE_1(R'_1, y_a), CE_1(R'_2, y_a), \dots, CE_1(R'_u, y_a))) \text{ with solution } b^{-1}x^{-1}. \\ &((S'_1, S'_2, \dots, S'_u), (CE_2(S'_1, y_b), CE_2(S'_2, y_b), \dots, CE_2(S'_u, y_b))) \text{ with solution } d^{-1}x^{-1}. \end{aligned}$$

We now give applications of our theorem for the protocols [10], [2], [6], [7], [11] there are algorithm that solve the MSCSP with non-negligible probability such as a length attack [5], so a length attack may be used for the protocol [10], [2], [6], [7]. Following the notation in [7], where G is a group so the security of the protocol in [7] is always based on the MSCSP (because we know the generators for the elements a_2 and b_1 we can use a length attack so disproving the claim in [7]), we have

$$\begin{aligned} CE_1(A_I, P_A) &= P_A^{-1} A_I P_A = a_2^{-1} w^{-1} a_1^{-1} A_I a_1 w a_2 = a_2 w^{-1} A_I w a_2, \\ A_I &\in A, A \text{ is a subgroup of } C_G(a_1) \\ CE_2(B_I, P_B) &= P_B B_I P_B^{-1} = b_1 w b_2 B_I b_2^{-1} w^{-1} b_1^{-1} = b_1 w B_I w^{-1} b_1^{-1}, \\ B_I &\in B, B \text{ is a subgroup of } C_G(b_2) \end{aligned}$$

if it may be easy for a some sets $\{g_1, g_2, \dots, g_k\}$ of the elements of G to compute a part of or all (if G is the braid group there are algorithms that will compute

a large part of the centraliser)

$$C(g_1, \dots, g_k) = C(g_1) \cap \dots C(g_k)$$

then we can compute the following

$$\begin{aligned} CE_3(C_I, P_A) &= P_A C_I P_A^{-1} = a_1 w a_2 C_I a_2^{-1} w^{-1} a_1^{-1} = a_1 w A_I w^{-1} a_1^{-1}, \\ C_I &\in C, C \text{ a subgroup of } C_G(b_2) \\ CE_4(D_I, P_B) &= P_B^{-1} D_I P_B = b_2^{-1} w^{-1} b_1^{-1} D_I b_1 w b_2 = b_2^{-1} w^{-1} B_I w b_2, \\ D_I &\in D, D \text{ a subgroup of } C_G(a_1) \end{aligned}$$

Following the notation in [6] let the elements transmitted by Alice and Bob be invertible then

$$\begin{aligned} CE_1(E_I) &= b_1^{-1} w^{-1} a_1^{-1} E_I a_1 w b_1 = b_1 w^{-1} E_I w b_1, E_I \in B \\ CE_2(F_I) &= b_2 w a_2 F_I a_2^{-1} w^{-1} b_2^{-1} = b_1 w F_I w^{-1} b_1^{-1}, B_I \in A \end{aligned}$$

and so we can use this equation in a length attack. It may be asked how much better is a length attack using the conjugacy equations above for [7],[6] (and related protocols) compared to the known length attacks on [7],[6], and if the above equations can be used to improve the known length attacks (for example we may try using one and/or both equations above they can be used to decide peeling off the correct generator in combination with the algorithm that decides to peel of generators in an existing attack, an example would be if the above existing algorithm is unable (i.e. pick at random) to decide which is the correct generator to peel then peeling from CE_1 and /or CE_2 may be used to decide the correct generator). Following the notation in [11] we have

$$CE_1(G_I, c) = c G_I c^{-1} = a^r b^s G_I b^{-s} a^{-r} = a^r b^\alpha a^{-r}, G_I = b^\alpha, \text{ for some } \alpha \text{ chosen by attacker}$$

or using the suggestion of using the element e in [11] we have

$$\begin{aligned} CE_2(H_I, c) &= c G_I c^{-1} = a^r e b^s H_I b^{-s} e^{-1} a^{-r} = a^r (e b^\alpha e^{-1}) a^{-r}, H_I = b^\alpha, \\ &\text{for some } \alpha \text{ chosen by attacker.} \end{aligned}$$

Following the notation in [2] we have

$$\begin{aligned} CE_1(K_I, y_1) &= y_1 K_I y_1^{-1} = a x a^{-1} K_I a x^{-1} a^{-1} = a x K_I x^{-1} a^{-1}, K_I \in R B_r \\ CE_2(L_I, y_2) &= y_2 L_I y_2^{-1} = b x b^{-1} L_I b x^{-1} b^{-1} = b x L_I x^{-1} b^{-1}, L_I \in L B_l \\ CE_3(M_I, y_1) &= y_1^{-1} M_I y_1 = a x^{-1} a^{-1} M_I a x a^{-1} = a x^{-1} M_I x a^{-1}, K_I \in R B_r \\ CE_4(N_I, y_2) &= y_2^{-1} L_I y_2 = b x^{-1} b^{-1} N_I b x b^{-1} = b x^{-1} N_I x b^{-1}, L_I \in L B_l \end{aligned}$$

Following the notation in [13] we have

$$\begin{aligned}
CE_1(T_I, w_i) &= w_i T_I w_i^{-1} = y_{i-1} v_i y_i^{-1} K_I y_i v_i^{-1} y_{i-1}^{-1} = y_{i-1} v_i K_I v_i^{-1} y_{i-1}^{-1} \\
CE_1(T_I, w_i^{-1}) &= w_i^{-1} T_I w_i = y_i v_i^{-1} y_{i-1}^{-1} K_I y_{i-1} v_i y_i^{-1} = y_i v_i^{-1} K_I v_i y_i^{-1} \\
K_I &\text{ is chosen from the subgroup that generates the elements } x_j \\
U_I &\text{ is chosen from the subgroup that generates the elements } y_j \\
\text{So the secrets } y_1, \dots, y_k &\text{ can be recovered and hence } v_i = y_{i-1}^{-1} w_i y_i \\
CE_2(U_I, w) &= w U_I w^{-1} = x_0 v_1 x_1 v_2 \dots v_k x_k U_I x_k^{-1} v_k^{-1} \dots v_2^{-1} x_1^{-1} v_1^{-1} x_0^{-1} \\
&= x_0 v_1 x_1 v_2 \dots v_k U_I v_k^{-1} \dots v_2^{-1} x_1^{-1} v_1^{-1} x_0^{-1} \\
&\text{recovering } x_0 v_1 x_1 v_2 \dots v_k \text{ gives } x_k = (x_0 v_1 x_1 v_2 \dots v_k)^{-1} w \text{ similarly} \\
CE_2(U_I, w^{-1}) &= w^{-1} U_I w \text{ gives } x_0. \text{ Because all } v_i \text{ can be recovered} \\
&\text{so similarly repeating the attack above using} \\
CE_2(U_I, (x_0 v_1)^{-1} w x_k^{-1} v_k^{-1}) &\text{ for } x_1, x_{k-1} \text{ and similarly by repeating} \\
&\text{again all the } x_i \text{ can be recovered.}
\end{aligned}$$

Following the notation in [10] we have

$$\begin{aligned}
CE_1(P_I, c_1) &= c_1 P_I c_1^{-1} = a_1 x a_2 P_I a_2^{-1} x^{-1} a_1^{-1} = a_1 x P_I x^{-1} a_2^{-1}, P_I \in UB_r \\
CE_1(Q_I, y_1) &= c_2 Q_I c_2^{-1} = b_1 x b_2 Q_I b_2^{-1} x^{-1} b_1^{-1} = b_1 x Q_I x^{-1} b_1^{-1}, K_I \in LB_r
\end{aligned}$$

The CKLHC protocol in [10] was introduced as an improvement of the KLCHKP protocol which it is a generalisation/modification of but we have shown the CKLHC does not improve the security of the KLCHKP protocol as they are both based on the MSCSP. This means using the KLCHKP and CKLHC cryptosystems is no more secure than using the AAG (Anshel-Anshel-Goldfeld) scheme [1] in the connection that they can all be broken using by solving the MSCSP. Hence this means there is no need to use the CKLHC cryptosystem any longer. The theorem implies the Turing reduction of the DH-DP to the MSCSP (MSCSP \propto_T DH-DP) for the case when the above DH-DP have related solutions, clearly a conjugacy extractor can be feasibly computed- that is in polynomial time and polynomial space (for parameters used in the CKLHC cryptosystem) using a finite number of group operations. If the conjugacy extractor is not computable in polynomial time and polynomial space in the connection of breaking a cryptographic protocol then the above protocol may be secure from an attack by solving the MSCSP, we may consider a generalisation of the MSCSP in the above cryptographic protocol where G is a semigroup instead of a group. Note if we have one CE then we exactly have the Turing reduction of the DP to the MSCSP (MSCSP \propto_T DP), and hence the Turing reduction of the DH-DP to the MSCSP (MSCSP \propto_T DH-DP).

In [14] an authentication scheme is given based on the problem of shifted conjugacy search problem (SCSP). It is not stated in [14] not to select r (the random value used in the commitment $x = r * p = r \cdot dp \cdot \sigma_1 \cdot dr^{-1}$) from a publicly known subgroup. Then an attack is as follows.

1. Suppose $r \in R$ where $R = \{\alpha_1, \dots, \alpha_k\}$ is a publicly known subgroup of B_n . In this step it is required the attacker just needs to find one element that commutes with r and not at least with $dp \cdot \sigma_1$ (using a chosen algorithm by the attacker) to show that SCSP can be reduced to solving the CSP. The attacker picks a subgroup of R given by the generators g_1, \dots, g_k . Then the attacker computes all of or a large part of

$$N = C(\alpha_1, \dots, \alpha_k) = C(\alpha_1) \cap \dots C(\alpha_k).$$

2. Then

$$\begin{aligned} CE_1(N_I, r * p) &= (r \cdot dp \cdot \sigma_1 \cdot dr^{-1})^{-1} N_I (r \cdot dp \cdot \sigma_1 \cdot dr^{-1}) = \\ &= dr \cdot \sigma_1^{-1} \cdot dp^{-1} \cdot r^{-1} \cdot N_I \cdot (r \cdot dp \cdot \sigma_1 \cdot dr^{-1}) = dr \cdot \sigma_1^{-1} \cdot dp^{-1} N_I dp \cdot \sigma_1 \cdot dr^{-1} \end{aligned}$$

will be true. $N_I \in N$, $1 \leq I \leq M$. The the protocol can be based on the MSCSP with

$$\begin{aligned} &((N_1, N_2, \dots, N_M), (CE(N_1, r * p), CE(N_2, r * p), \dots, CE(N_M, r * p))) \\ &\text{with solution } (dr \cdot \sigma_1^{-1} \cdot dp^{-1}, dp \cdot \sigma_1 \cdot dr^{-1}), O = dp \cdot \sigma_1 \cdot dr^{-1}. \end{aligned}$$

and r can be found by computing $(\sigma_1^{-1} \cdot dp^{-1} O)^{-1} = r$, there is a similar attack if s (Alice's private key) is chosen from a subgroup that is publicly known. Note the similar attack with N_I commuting $r \cdot dp$ would mean the SCSP is just the CSP (no extra computation using d is required).

As a variant of the above algorithm an attacker may try to compute an element $N'_I \in C(L)$ then it may be possible to use N'_I instead of N_I in the attack above where $L = r * p$ or $L = r * p'$, so in this variant knowledge of s ($L = p'$ here) or r being chosen from a subgroup is not required. A different second CE on the authentication scheme in [14] or the SCSP, is suppose in a general case we have a pair of examples of the SCSP that have the same secret element $x = r * p, x' = r * p'$ (the notation here follows [14] with the secret element r) then

$$\begin{aligned} CE_2(x, x'^{-1}) &= \\ CE_2(r * p, r * p') &= x \cdot x'^{-1} \\ &= (r \cdot dp \cdot \sigma_1 \cdot dr^{-1}) \cdot (r \cdot dp' \cdot \sigma_1 \cdot dr^{-1})^{-1} \\ &= (r \cdot dp \cdot \sigma_1 \cdot dr^{-1}) \cdot dr \cdot \sigma_1^{-1} \cdot dp'^{-1} \cdot r^{-1} \\ &= r \cdot dp \cdot dp'^{-1} \cdot r^{-1} \end{aligned}$$

so the secret r can be found by solving the CSP pair $(dp \cdot dp'^{-1}, CE(r * p, r * p'))$ for r , there is a similar CE for dr (use $x^{-1} \cdot x'$ instead of $x \cdot x'^{-1}$ and then dr is transformed to r etc). Then the attack waits until $b = 1$ is so in this case Alice send to Bob $r * s = r \cdot ds \cdot \sigma_1 \cdot dr^{-1}$ hence the attacker computes the private key $s = r^{-1} \cdot (r * s) \cdot dr \cdot \sigma_1^{-1}$. We note the above attacks is can be used to answer question 2.6 in [14] (with $y = p$ in th CSP). We note our attack can be

used to solve the shifted conjugacy decision problem. Our results suggest that CE functions should be hard to compute hence semigroups may be considered because then the theorem 1 may be false because the elements y_a and/or y_b are not invertible. This suggestion applies to any hard problem such as the EDL problem below. Note in the algorithms in CE computations it may be centraliser element(s) (call these p_i) that multiply the secret(s) cancel out and it can be shown these factors in the centraliser are efficiently computable, for example one way to do this is if p_i is a power of the fundamental braid then we can estimate a power of the fundamental braid from the public elements (for example using a length function) and so recover a p_i , or instead find this power by brute force.

We now consider a problem related to the SCSP/CSP which is given a semigroup G , and publicly known functions $u : G \rightarrow G_1, v : G_2 \rightarrow G, w : G_3 \rightarrow G$, G_1, G_2, G_3 are subsets of G , and publicly the publicly known elements $y_i = u(r)v(p_i)w(r^{-1}), v(p_i), 1 \leq i \leq n$ find if the element r . We observe that the problem generalises the twisted conjugacy problem [15] and the doubly twisted conjugacy problem [15], e.g. with $i = 1, u$ an endomorphism, v, w the identity map we recover the twisted conjugacy problem, G a group, $G_1 = G, G_2 = G, G_3 = G$; so we refer to the above problem as the GTCP (generalised twisted conjugacy problem) which we now describe solutions for. Now consider the GTCP with $i > 1$, select a pair i, j with $v(p_i) \neq v(p_j), 1 \leq i, j \leq n$ we have the conjugacy extractor

$$\begin{aligned} CE_1(y_i, y_j) &= y_i \cdot y_j^{-1} \\ CE_1(u(r)v(p_i)w(r^{-1}), u(r)v(p_j)w(r^{-1})) &= u(r)v(p_i)w(r^{-1}) \cdot (u(r)v(p_j)w(r^{-1}))^{-1} \\ &= u(r)v(p_i)w(r^{-1}) \cdot w(r^{-1})^{-1}v(p_j)^{-1}u(r)^{-1} \\ &= u(r)v(p_i)v(p_j)^{-1}u(r)^{-1} \end{aligned}$$

so here we solve the CSP pair $(v(p_i)v(p_j)^{-1}, CE_1(y_i, y_j))$. Once $u(r)$ is obtained (attempt to) use the inverse of u to get r .

$$\begin{aligned} CE_2(y_i, y_j) &= y_j^{-1} \cdot y_i \\ CE_2(u(r)v(p_i)w(r^{-1}), u(r)v(p_j)w(r^{-1})) &= (u(r)v(p_j)w(r^{-1}))^{-1} \cdot u(r)v(p_i)w(r^{-1}) \\ &= w(r^{-1})^{-1}v(p_j)^{-1}u(r)^{-1} \cdot u(r)v(p_i)w(r^{-1}) \\ &= w(r^{-1})^{-1}v(p_j)^{-1}v(p_i)w(r^{-1}) \end{aligned}$$

so here we solve the CSP pair $(v(p_j)^{-1}v(p_i), CE_2(y_i, y_j))$. Once $w(r)$ is obtained use the inverse of w to get r . Note for one of the above CE functions for the twisted conjugacy problem the problem is just the conjugacy search problem. The above can be repeated for different i, j to get a reduction to the MSCSP. Another method to solve the GTCP as follows.

1. Suppose $G_1 = \{\alpha_1, \dots, \alpha_k\}, G_3 = \{\beta_1, \dots, \beta_l\}$ are publicly known. In this step it is required that one element that commutes with r is to be found. Pick subgroups of G_1, G_3 given by the generators $g_1, \dots, g_L, h_1, \dots, h_L$. Then compute

all of or a large part of

$$\begin{aligned} N &= C(\alpha_1, \dots, \alpha_k) = C(\alpha_1) \cap \dots C(\alpha_k). \\ M &= C(\beta_1, \dots, \beta_k) = C(\beta_1) \cap \dots C(\beta_l). \end{aligned}$$

2. Then

$$\begin{aligned} CE_3(M_I, y_i) &= u(r)v(p_i)w(r^{-1})M_I(u(r)v(p_i)w(r^{-1}))^{-1} = \\ &= u(r)v(p_i)w(r^{-1})M_Iw(r^{-1})^{-1}v(p_i)^{-1}u(r)^{-1} = \\ &= u(r)v(p_i)M_Iv(p_i)^{-1}u(r)^{-1} \end{aligned}$$

and

$$\begin{aligned} CE_4(N_I, y_i) &= (u(r)v(p_i)w(r^{-1}))^{-1}N_Iu(r)v(p_i)w(r^{-1}) = \\ &= w(r^{-1})^{-1}v(p_j)^{-1}N_Iv(p_i)w(r^{-1}) \end{aligned}$$

$N_J \in N, M_I \in M, 1 \leq I \leq m, 1 \leq J \leq n$. Hence using $CE_3(M_I, y_i), CE_4(N_I, y_i)$: the GTCP has a solution respectively in the MSCSPs with

$$\begin{aligned} &((N_1, N_2, \dots, N_n), (CE_4(N_1, y_{j_1}), CE_4(N_2, y_{j_2}), \dots, CE_4(N_n, y_{j_n}))) \\ &\text{with solution } w(r^{-1})^{-1}v(p_j)^{-1}; \end{aligned}$$

$$\begin{aligned} &((M_1, M_2, \dots, M_m), (CE_3(M_1, y_{i_1}), CE_3(M_3, y_{i_2}), \dots, CE_3(M_m, y_{i_m}))) \\ &\text{with solution } u(r)v(p_i); \end{aligned}$$

; from $w(r^{-1})^{-1}v(p_j)^{-1}, u(r)v(p_i)$ we can obtain r by using a right multiplication and using the inverses of w, u ; this show the twisted conjugacy problem can be deterministically reduced to the MSCSP. We observe the algorithm in section 7 below can be used to attempt to solve the twisted or doubly twisted conjugacy problem with or without using u or v . Observe once we have found a solutions to the twisted conjugacy problem, SCSP this means we can solve the decision version of the twisted conjugacy problem and SCSP.

6. A Solution for the EDL type problem in Non-commutative Semigroups.

The EDL braid type problem was proposed in [9] where it is assumed to be hard. Following the notation in our definition above of the EDL we have the following theorem.

Theorem 2

Given (the DP equations) $y_a = uav, y_b = wbx$ in the EDL is sometimes equivalent to solving the CSP if y_a, y_b are both invertible elements.

Proof

We will solve this problem by solving a system of DP equations for certain values of the secrets so our proof can be used to solve a system of DP equations for example in showing above the shifted conjugacy based protocol is based on

the CSP. Assume $w = u$ and $x = v, a \neq b$, then $y_a = uav$, $y_b = ubv$, and so the CE functions give

$$\begin{aligned} y_a y_b^{-1} &= uab^{-1}u^{-1} \\ y_a^{-1} y_b &= v^{-1}a^{-1}bv \end{aligned}$$

and so we can solve the CSP pairs for $(ab^{-1}, y_a y_b^{-1})$, and $(a^{-1}b, y_a^{-1} y_b)$ for $g_1 = u$ and $g_2 = v$ respectively using an algorithm for the CSP. The verification (can be done efficiently using the algorithms for the word problem when G is the braid group e.g. see [10]) $y_a = g_1 a g_2$, $y_b = g_1 b g_2$ will be true by the above assumption, hence we have shown the EDL to be true in this case. For some examples of the CSP there are fast algorithm for it for example see [4] hence the assumption in [9] that the EDL is hard is not always true. If we know the generators of the subgroups A and B then we may use a length based algorithm to recover the secret element with non-negligible probability.

We re-define again the EDL problem more generally as follows

Public information: G is a Semigroup. A, B, C, D are subsets of G . $a_i, b_i, x_i, y_i \in G$ with $y_i = a_i x_i b_i$, $1 \leq i \leq m$

Secret Information: $a_i \in A_i$, $b_i \in B_i$, (A_i and B_i are subgroups).

Objective: Decide if $F_{x_1}(y_1) \cap F_{x_2}(y_2) \dots \cap F_{x_r}(y_r) \neq \emptyset$. Where $F_\beta(\alpha) = \{(a, b) \in B_n \times B_n : \alpha = a\beta b\}$.

Theorem 3

Given the generalised EDL above is sometimes equivalent to solving the CSP. The generalised EDL may be partially solved in the connection a subset integers t_1, t_2, \dots, t_r in $[1, m]$ we can decide if $F_{x_{t_1}}(y_{t_1}) \cap F_{x_{t_2}}(y_{t_2}) \dots \cap F_{x_{t_r}}(y_{t_r}) \neq \emptyset$.

Proof

Again we will solve this problem by solving the a system of DP equations for certain values of the secrets so our proof can be used to solve a system of DP equations. Assume $a_i = a_j$ and $b_i = b_j$ for all $i, j \in [1, m]$ and $i \neq j$ then $y_i = a_i x_i b_i$, $y_j = a_j x_j b_j$, and so the CE functions give

$$\begin{aligned} y_i y_j^{-1} &= a_i x_i x_j^{-1} a_i^{-1} \\ y_i^{-1} y_j &= b_i^{-1} x_i^{-1} x_j b_i \end{aligned}$$

and so we can solve the CSP pairs $(x_i x_j^{-1}, y_i y_j^{-1})$, and $(x_i^{-1} x_j, y_i^{-1} y_j)$ for the solutions $g_1 = a_i$ and $g_2 = b_i$ respectively using an algorithm for the CSP. We can get more conjugacy extractor functions by choosing different values for i and j . The verification (can be done efficiently using the algorithms for the word problem when G is the braid group e.g. see [10]) $y_a = ? g_1 x_a g_2$, $y_b = ? g_1 x_b g_2$ will be true by the above assumption for all $(a, b) = (i, j)$, hence we have shown the EDL to be true in this case.

If we know the generators of the subgroups A and B then we may use a length based algorithm to recover the secret element with non-negligible probability.

The EDL can be partially solved if it is true that the assumption $a_i = a_j$ and $b_i = b_j$ for at least two integers i and j , $i, j \in [1, m]$ and $i \neq j$. Then the

proof that the EDL can be partially solved is the same as above except there are fewer choices for i and j .

7. Second Algorithm using CE Functions

Given (the DP equation) $u = xaz$. This attack reveals partial information about the secret z or totally recover z . This attack is a generalisation of our attack on the DP by using a MSCSP.

1. The attacker picks elements S_I according to some criteria relating to commutativity, for example elements S_I may be picked randomly or S_I may be composed of a few Artin generators as these may commute to some degree with z .

2. Then for $1 \leq I \leq M$ for a sequence of integers T_I

$$CE_I(S_I, u) = uS_Iu^{-1} = xazS_Iz^{-1}a^{-1}x^{-1} = xa\bar{z}S_I\bar{z}^{-1}a^{-1}x^{-1}$$

where (with probability ρ) \bar{z} is a partial factor of z for some I this means $z = z_{T_I}\bar{z}_{T_I}$.

3. We solve for each I the CSP $(S_I, CE_I(S_I, u))$ for the solution $xa\bar{z}$ and hence compute $z_{T_I} = ((xa\bar{z}^{-1})^{-1}xaz)^{-1}$. Note if S_I is selected from the centraliser of z then we can use the MSCSP at this step (so this shows DP is Turing reducible to MSCSP).

4. We now find (in some way) z using the information $(S_I, xa\bar{z}, z_{T_I})$ and the other information used in the protocol. One of the simplest choices to implement this step is trying to find \bar{z} for each I by brute force and hence possibly recover z .

A variant of the above attack is after z_{T_I} is recovered is to repeat at the attack (at least once) by iterating with $uz_{T_I}^{-1}$ instead of u (and obviously all other values may be different) so in this way we may be able to find a bigger factor of z . It may be true (with some probability ρ_2) that \bar{z} contains a partial factor of a which means the CSP is solved to give $\bar{z}_{T_I}a_{T_I}$ where a_{T_I} is some partial factor of a . Then the simplest choice at this step to recover z is to find a_{T_I} by brute force and use a_{T_I} to recover z .

Conclusion

We have shown the protocols [10],[2],[6],[7],[11],[13],[14] can have security based on the MSCSP. We have shown the DP and DH-DP can be solved by the MSCSP. Our theorem 1 implies that the CKLHC cryptosystem and related cryptosystems are MSCSP based so are no more secure than using the AAG protocol [1]. Our theorem 1 implies that semigroups should (for G) be used for the protocol in [12] to be secure so not based on the MSCSP. We should not use the CKLHC protocol in [10] or related protocols (which are suggested to be used in braid groups) compared to using the AAG protocol as it is no more secure than using the AAG in the connection they are based on the MSCSP.

References

- [1] I. Anshel, M. Anshel, D. Goldfeld. An Algebraic Method for Public-key Cryptography, Mathematical Research Letters, 6, 1999, pp. 287-292.
- [2] K.K. Ko, S.J. Lee, J.H. Cheon, J.W. Han, J.S. Kang, C. Park. New public-key cryptosystem using braid groups. CRYPTO 2000. LNCS, 1880, 2000, pp. 166-183.
- [3] K. H. Ko, Tutorial on Braid Cryptosystems 3, PKC 2001, Korea, February 13-15, 2001. Available at www.ipkc.org/pre_conf/pkc2001/PKCtp_ko.ps
- [4] V. Gebhardt. A New approach to the conjugacy problem in Garside groups. J Algebra 292 (1) , 2005, pp. 282-302.
- [5] D.Garber, S. Kaplan, M. Teicher, B. Tsaban, U. Vishne. Length-Based Conjugacy Search in the Braid Group, available at <http://arxiv.org/abs/math.GR/0209267>
- [6] V. Shpilrain and A. Ushakov, "Thompson's group and public key cryptography", LNCS, 3531, Springer-Verlag, 2005, pp. 151-164.
- [7] V. Shpilrain and A. Ushakov, A new key exchange protocol based on the decomposition problem, available at <http://eprint.iacr.org/2005/447.pdf> (Accessed 20/05/2006)
- [8] J. Cheon and B. Jun, "A Polynomial Time Algorithm for the Braid Diffie Hellman Conjugacy Problem", *Proceedings of Crypto 2003*, LNCS, 2729, Springer-Verlag , 2003, pp. 212-215.
- [9] T. Thomas and A. K. Lal, Group Signature Schemes Using Braid Groups, arXiv:cs.CR/0602063, 2006
- [10] J.C. Cha, K.H. Ko, S.J. Lee, J.W. Han, J.H. Cheon, An efficient implementation of braid groups, Advances in cryptology: Proceedings of ASIACRYPT 2001, Lecture Notes in Computer Science, Springer-Verlag, 2248, pp.144-156, 2001
- [11] E. Stickel, A New Method for Exchanging Secrets, Proceedings of the Third International Conference on Information Technology and Applications, 2005
- [12] M. M. Chowdhury, Key Agreement and Authentication Scheme Using Non-Commutative Semigroups, ArXiv preprint, August 2007
- [13] Tang Xue-Ming, Hong Fan, Cui Guo-Hua, A Public Key Encryption Algorithm on Braid Groups, Journal of Software, Vol 18, No. 3, March 2007, pp.722-729
- [14] P. Dehornoy, Using Shifted Conjugacy in braid-based cryptography, arXiv preprint, 2006

- [15] P. Staecker, Computing Twisted Conjugacy Classes for Free groups using nilpotent quotients, 2007